Lep-MAP3 : robust linkage mapping even for low-coverage whole genome sequencing data

Rastas, Pasi

2017-12-01

Rastas , P 2017 , ' Lep-MAP3 : robust linkage mapping even for low-coverage whole genome sequencing data ' , Bioinformatics , vol. 33 , no. 23 , pp. 3726-3732 . https://doi.org/10.1093/bioinformatics/b

http://hdl.handle.net/10138/312091
https://doi.org/10.1093/bioinformatics/btx494

acceptedVersion

Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Subject Section

Lep-MAP3: Robust linkage mapping even for low-coverage whole genome sequencing data

Pasi Rastas (pasi.rastas@helsinki.fi)

Department of Zoology, Butterfly Genetics Group, University of Cambridge, UK
Department of Biosciences, Ecological Genetics Research Unit, University of Helsinki, Finland

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Accurate and dense linkage maps are useful in family-based linkage and association studies, quantitative trait locus mapping, analysis of genome synteny and other genomic data analyses. Moreover, linkage mapping is one of the best ways to detect errors in *de novo* genome assemblies, as well as to orient and place assembly contigs within chromosomes. A small mapping cross of tens of individuals will detect many errors where distant parts of the genome are erroneously joined together. With more individuals and markers, even more local errors can be detected and more contigs can be oriented. However, the tools that are currently available for constructing linkage maps are not well suited for large, possible low-coverage, whole genome sequencing datasets.

Results: Here we present a linkage mapping software Lep-MAP3, capable of mapping high-throughput whole genome sequencing datasets. Such data allows cost-efficient genotyping of millions of single nucleotide polymorphisms (SNPs) for thousands of individual samples, enabling, among other analyses, comprehensive validation and refinement of *de novo* genome assemblies. The algorithms of Lep-MAP3 can analyse low-coverage datasets and reduce data filtering and curation on any data. This yields more markers in the final maps with less manual work even on problematic datasets.

We demonstrate that Lep-MAP3 obtains very good performance already on 5x sequencing coverage and outperforms the fastest available software on simulated data on accuracy and often on speed. We also construct *de novo* linkage maps on 7-12x whole-genome data on the Red postman butterfly (*Heliconius erato*) with almost 3 million markers.

Availability: Lep-MAP3 is available with the source code under GNU general public license from http://sourceforge.net/projects/lep-map3.

Contact: pasi.rastas@helsinki.fi

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

High-throughput sequencing and computational advances have enabled practical ways to assemble genome sequences *de novo* (Simpson and Pop, 2015). However, typically (*de novo*) assemblies contain assembly errors and are fragmented in many short contigs (or scaffolds), making it difficult to know how the sequences are physically located with respect to each other (Fierst, 2015; Simpson and Pop, 2015). When applicable, linkage mapping provides a practical way to anchor and orient the sequences into chromosomes (Fierst, 2015). This information can directly be used for scaffolding contigs into longer sequences or enable local reassembly given the approximate locations and partial orientations of the sequences. There are some software for integrating assemblies and linkage maps, like ArkMAP (Paterson and Law, 2013) and Chromonomer (http://catchenlab.life.illinois.edu/chromonomer/, Catchen (2015)).

The number of individuals (offspring) in a mapping cross defines how many recombinations can be detected. To orient a contig, there must be at least two markers in that contig and at least one recombination between those markers. Moreover, each recombination can orient at most one contig. Even a mapping cross of tens of individuals can detect many assembly errors where distant parts are erroneously joined together (Rastas *et al.*, 2013). With more individuals, even more local errors can be detected

and more contigs can be oriented and placed into chromosomes. As well as the number of individuals, the number of markers affects the map resolution. With too few markers, shorter contigs remain without any/proper linkage information, and some recombinations will be missed which reduces information on the orientation.

Low-coverage high-throughput whole genome sequencing has large potential in linkage mapping. It cost-efficiently obtains genotype information for millions of single nucleotide polymorphisms (SNPs) and thousands of individuals even for non-model species, enabling to pinpoint most recombinations within narrow regions in the genome. However, the tools that are currently available for constructing linkage maps are not well suited for this many markers and even less so for low to medium coverage sequencing data.

Dense high-quality linkage maps are useful, as well as required, for family-based linkage and association analysis (Laird and Lange, 2008), quantitative trait locus (QTL) mapping (Doerge, 2002), analysis of genome synteny and other genomic data analyses.

1.1 Previous work

Linkage map construction is well studied and a fundamental computational problem in genetics. Most available software are listed in the review (Cheema and Dicks, 2009), some notable software being CRI-MAP (Lander and Green, 1987), JoinMap (Van Ooijen, 2011) and MSTmap (Wu *et al.*, 2008). More recent software since this review include Lep-MAP (Rastas *et al.*, 2013), High-MAP (Liu *et al.*, 2014) and Lep-MAP2 (Rastas *et al.*, 2016).

In this article, we present a novel linkage mapping software Lep-MAP3 (LM3), capable incorporating all potential markers from whole genome sequencing, while being equally useful on smaller dataset obtained from, e.g. RAD-sequencing. Its most novel feature is that it accepts and makes use of data as genotype likelihoods. This allows LM3 to obtain information on genotype uncertainty and enables linkage mapping on low-coverage sequencing data.

It also reduces mapping errors by modelling recombination interference and scales, in speed and modelling accuracy, to much larger datasets than was possible with the existing software. LM3 is memory efficient and automated, and can use simultaneously data on multiple full-sib families (or crosses that can be analysed as such, e.g. F2 crosses, half-sibs, doubled haploid, backcross, RIL). Finally, it can take into account achiasmatic meiosis, a special feature of Lepidoptera and some other taxa with recombination only in one sex.

Previously, the performance of many linkage mapping software has been compared against each other. In Rastas *et al.* (2016), Lep-MAP2, Lep-MAP, TMAP, JoinMap and HighMap were compared and Lep-MAP2 was found to be superior among compared software. Software MSTmap is known to be very fast (Fierst, 2015), compared for example in Rastas *et al.* (2013), and to our knowledge, MSTmap is the only available software capable of mapping over 10,000 markers in a reasonable time (under an hour). However, MSTmap has some limitations, for instance being restricted to only single families and phased data, whereas LM3 does not have such limitations. To allow comparisons between MSTmap and LM3 in this article, we used simulated double haploid (DH) data given in phase-known format.

1.2 Differences between LM3 and its previous versions

LM3 is based on a similar philosophy to it's earlier versions LM1 (Lep-MAP1) (Rastas *et al.*, 2013) and LM2 (Lep-MAP2) (Rastas *et al.*, 2016). The main difference between LM3 and LM1&2 is that the input genotype likelihoods are used in each step of the map construction.

The modules are named similarly as in LM2 with number 2 added if the same name was used in LM2, e.g. ParentCall $\mapsto$ ParentCall2. The usage of each module is similar as in LM2, thus if you have previously used LM2, you can easily run the same commands with LM3, just by producing (or converting) the data in the new format.

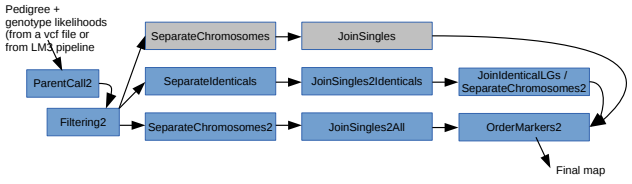


Fig. 1: Typical data processing pipeline with LM3 from the input genotype likelihoods to the final map. Three alternative ways are described, where the upmost path uses Lep-MAP2 modules.

The module ParentCall2, now supporting vcf files and grandparental information, is an improved version of ParentCall in LM2 which is an improved version of Counts2Genotypes found already in LM1. Other main modules are mostly new, using similar hierarchy for inner classes as in LM2. Code for handling input and output are mostly reusing earlier code. The data processing pipeline is similar to one used in Rastas *et al.* (2016), derived from the pipelines used with LM1.

The marker ordering (OrderMarkers2) is now now more robust to noisy data and up to 450x-2000x faster (on simulated data, see the section 3). It can also natively use genotypic information on grandparents to phase the data accordingly, which eases QTL mapping. Finally, the underlying haplotype model has changed (see Figure 2).

2 Methods

The LM3 (Lep-MAP3) workflow is illustrated in Figure 1. This workflow consists of modules ParentCall2, Filtering2, SeparateChromosomes2 (SeparateIdenticals), JoinSingles2All (JoinSingles2Indenticals), JoinIndenticalLGs and OrderMarkers2. Alternatively, Lep-MAP2 modules SeparateChromosomes and JoinSingles (upmost row in Figure 1) can be used before OrderMarkers2, providing faster alternative when genotypes can be called with good quality.

The genetic input data for all modules consists of a pedigree describing one or several full-sib families and genotype likelihoods for each marker and individual. The first step of LM3, the ParentCall2 module will call parental genotypes by taking into account genotype information on parents and offspring (and grandparents if they are present).

The Filtering2 allows user to filter markers based on, e.g. segregation distortion and amount of missing data. SeparateIdenticals and SeparateChromosomes2 clusters (and separates) markers by calculating LOD scores between all pairs of markers, the difference being that SeparateIdenticals only clusters markers that segregate exactly identically (recombination rate $\theta = 0$) while SeparateChromosomes2 clusters actual chromosomes or linkage groups. Modules JoinSingles2Identicals and JoinSingles2All will add additional markers to the found marker groups.

JoinIdenticalLGs can be used to cluster found identical marker groups to chromosomal (linkage) groups. The clustering of identical markers reduces the required time on whole genome data where the number of markers can be orders of magnitude larger than the number of differently segregating markers.

Finally, the markers separated into linkage groups can be ordered using OrderMarkers2 module. This marker ordering step is the main computational step in linkage mapping.

Notation. We consider only markers where at least one parent is (recombination) informative, i.e. heterozygous. We define an (informative) haplotype as alleles inherited from informative parent, i.e. maternal or

	father	mother	child	haplotype	
only mother informative:	AA	AB	AA	00 or 10	(= ?0)
	AA	AB	AB	01 or 11	(= ?1)
only father informative:	AB	AA	AA	00 or 01	(= 0?)
	AB	AA	AB	10 or 11	(= 1?)
both parents informative:	AB	AB	AA	00	
	AB	AB	AB	01 or 10	
	AB	AB	BB	11	

Table 1. Genotypes and haplotypes for genetic marker with alleles A and B. Haplotype alleles are 0 and 1 (? is an unknown allele).

paternal allele arbitrary denoted as 0 and 1. In a phased haplotype, alleles are mapped so that a change in individual's haplotype ($0 \rightarrow 1$ or $1 \rightarrow 0$) in the marker order indicates recombination or genotyping error. We do not require grandparental phase, i.e. inheritance vectors, as we can detect recombinations in parental phase. However, LM3 accepts phased data as input but does not support mixing of phased and unphased data (unless all unphased).

By genotype likelihoods we mean probability $P(d|g)$ = probability of the data d given the genotype g. Such likelihood can be obtained from sequencing or SNP-assay based data. From the genotype likelihoods of a offspring and its parents, we can infer four values, $p_{00}, p_{01}, p_{10}, p_{11}$, giving the probabilities for informative haplotypes 00,01,10 and 11, where the first digit is the paternal and the second digit is the maternal haplotype. Examples of genotype combinations and the corresponding haplotypes are given in Table 1.

2.1 Clustering markers into chromosomes

LM3 separates chromosomes (or linkage groups) by evaluating two-point LOD scores (Morton, 1955) between markers. The novelty of LM3 is that all computations are carried using genotype likelihoods, instead of genotypes. In the next subsection, the LOD score computation is explained and the marker clustering modules are sketched briefly.

2.1.1 Computing LOD scores

Let the haplotype probabilities for two markers be p_* and q_*, and the recombination rates be θ_1 and θ_2 for male and female, respectively.

The probability that paternal haplotypes are identical between these two markers is $a = (p_{00}+p_{01})(q_{00}+q_{01})+(p_{11}+p_{10})(q_{11}+q_{10})$, and similarly $b = (p_{00}+p_{10})(q_{00}+q_{10})+(p_{11}+p_{01})(q_{11}+q_{01})$ for maternal haplotypes. If these two markers are in identical phase, the LOD score contribution ($\log(P(G|\theta=[\theta_1,\theta_2])/P(G|\theta=0.5))$, where G is the genotype data and θ is the recombination rate) is

$$\log\Big(\frac{((1-\theta_1)a+\theta_1(1-a))\cdot((1-\theta_2)b+\theta_2(1-b))}{1/4}\Big). \quad (1)$$

A key observation here is that this equation works well with $\theta = 0$, thus it can be used to find identical markers or for species without recombination in one sex. The parameters θ_1 and θ_2 are user defined in LM3. To handle phase unknown data, all 4 possible phases are tested and the one giving highest LOD score is used.

2.1.2 Separating identical markers

SeprateChromosomes2 module in LM3 evaluates LOD scores for all pairs of markers and joins markers to form linkage groups. This is practical for smaller datasets, but not for whole genome datasets due to its quadratic time dependency on the number of markers. However, the SeparateIdenticals module is much faster as it collapses identical markers and can divide data into k (given by the user) independent parts. Dividing data into k parts yields a speedup of k (but can miss rare markers). Moreover, as the parts are independent, their computations can be executed in parallel on multiple cores. Finally the collapsed identical markers for all parts are joined together. The marker separating modules have been earlier used and briefly explained in Van Belleghem *et al.* (2017). Note that the separation of identical markers can be run very efficiently by running it separately for each contig or, say in 1Mb windows. The found identical markers can be collapsed and used as "binned" markers to reduce the computational burden of linkage mapping, but we have found this process to be challenging in practise. Moreover, LM3 can be used to collapse identical marker, e.g. by using the physical locations of the markers, whilst the preferred way is to treat markers individually.

2.2 Ordering markers

LM3 orders markers by maximising the likelihood of the data. Next the used model and algorithms are sketched.

2.2.1 Haplotype model

The phased haplotypes are modelled with a hidden Markov model, illustrated in Figure 2 (b). The model has only two parameters θ_s and i_s for both sexes $s = 1, 2$, where θ_s defines the recombination probability and the i_s is the probability of recurrent recombination (interference). These parameters are given by the user, thus there is no computational overhead on learning these parameters (unlike in Lep-MAP2 model, Figure 2 (a)). The haplotype likelihoods (see section 2.1.1) define the emission probabilities naturally. The likelihood of data is computed by Viterbi type algorithm, i.e. the likelihood is defined by the two paths (maternal and paternal) through the haplotype model that give highest probability for the data. The likelihood can be evaluated in $O(mn)$ time, where n is the number of markers and m is the number of individuals. In the same time the marker positions (in centiMorgans) are obtained by keeping track of the two paths, each recombination corresponds taking path up ($\nearrow$) in the model.

The parameters i_s model the recombination interference in a simple way; on each individual the first recombination decreases likelihood by θ_s, while the following by $\theta_s i_s$. Setting $i_s = 1$ does not add any penalty for recurrent recombinations, but no recurrent recombinations supported by only a single marker will be allowed due to the topology of the model. Setting $i_s = 0$ does not allow any recurrent recombinations. Achiasmatic meiosis can be facilitated by setting $\theta_s = 0$ for $s = 1$ or $s = 2$.

2.2.2 Phasing

The phase of the data can be given by the user, but in the phase-unknown case, the phase is estimated using the haplotype model (Figure 2 (b)) every time the data likelihood (given the order) is evaluated. This is done as follows. The likelihood of a random (or previously found) phasing is first evaluated. Then it is evaluated whether changing phases for a single marker or for all markers right of this marker improves the likelihood. If the change improves likelihood, the phasing is changed accordingly. The phase changes and updates for all markers (from right to left) can be computed in $O(mn)$ time and often 1-2 runs of this phasing step are sufficient to find and verify (locally) optimal phasing. Moreover, typically the phasing does not change after the first phasing steps.

2.2.3 Marker ordering algorithm

The divide-and-conquer algorithm for ordering markers is sketched in Algorithm 1 and graphically illustrated in Figure 3. It is iterated a user defined times, starting from a random marker order or from a order given by the user, and after each iteration, the order is polished by POLISH and by local changes in each window of five adjacent markers.

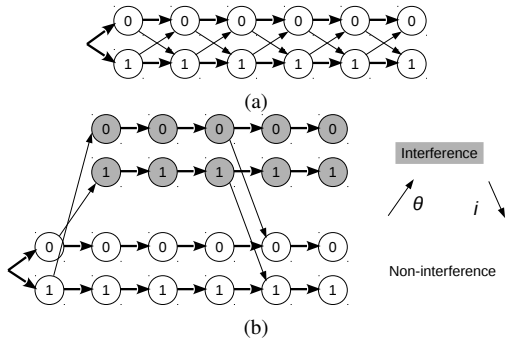


Fig. 2: The Lep-MAP2 model (a) and the LM3 model (b) for phased haplotypes over six markers (stacked states). The zeros and ones in the states correspond to haplotypes and a transition between states of different haplotypes models recombination. For clarity, only two interference (↘) and two recombination transitions (↗) are illustrated in the LM3 model.

Algorithm 1 Marker Ordering

procedure ORDER(markers $M = m_1, m_2, \ldots, m_k$)
  if $k \leq 2$ **return** M
  Divide markers M into two equal size parts M_1 and M_2
  $M_1 :=$ ORDER(M_1)
  $M_2 :=$ ORDER(M_2)
  $R_1 :=$ POLISH(MERGE(M_1, M_2)))
  $R_2 :=$ POLISH(MERGE(M_1, REVERSE(M_2)))
  if likelihood(R_1) >likelihood(R_2) **then**
    return R_1
  else
    return R_2

procedure REVERSE(markers $M = m_1, m_2, \ldots, m_k$)
  return $m_k, m_{k-1}, \ldots, m_1$

procedure MERGE(two lists of markers $M_1 = m_{11}, \ldots, m_{1|M_1|}$ and $M_2 = m_{21}, \ldots, m_{2|M_2|}$)
  $S(0,0) = 0$ ▷ $S(i,j)$ is the max score of the merging of $m_{21}, \ldots, m_{2j}$ and M_1 by adding m_{2j} before or just after m_{1i}
  for $j = 1$ to $|M_2|$ **do**
    $S(0,j) = S(0,j-1)+$SCORE($M_1, 0, m_{2j}$)
    for $i = 1$ to $|M_1|$ **do**
      $S(i,j) = \max\{S(i-1,j), S(i,j-1)+$ SCORE(M_1, i, m_{2j})$\}$
  Trace-back the path obtaining score $S(|M_1|,|M_2|)$ and add markers of M_2 between markers of M_1 correspondingly to obtain marker order R.
  return R

procedure SCORE(markers $M = m_1, m_2, \ldots, m_k$, integer i and marker n)
  return $\log\left(\frac{\text{likelihood}(m_1, m_2, \ldots, m_i, n, m_{i+1}, \ldots, m_k)}{\text{likelihood}(m_1, m_2, \ldots, m_k)}\right)$

procedure POLISH(markers $M = m_1, m_2, \ldots, m_k$)
  for $i = 0$ to k **do**
    $R_i := \emptyset$
  for $j = 1$ to k **do**
    $i := arg\max_i\{SCORE(M, i, m_j) : 0 \leq i \leq k\}$
    $R_i := R_i \cup \{m_j\}$
  return markers in order $R_0, R_1, \ldots, R_k$

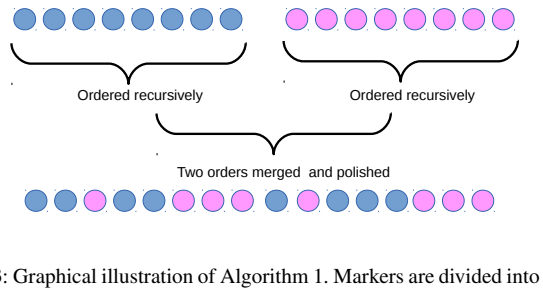


Fig. 3: Graphical illustration of Algorithm 1. Markers are divided into two parts which are recursively ordered following the merging (and polishing) of the two parts together.

2.3 Analysing and improving marker ordering runtime

During the execution of ORDER, the procedures MERGE and POLISH both evaluate SCORE $O(n^2)$ times. Assuming the phases of markers M are known (on unphased data, the phase is determined as described in section 2.2.2), the SCORE can be calculated in $O(m)$ time after $O(mn)$ preprocessing by typical forward-backward type dynamic programming algorithm. This procedure yields total runtime of $O(mn^2)$ and $O(mn)$ memory requirement.

This algorithm is very fast, and on our experiments we have been able to run about 20,000 markers on 100 individuals in about 1 hour on a single Desktop computer CPU. However, the algorithm's quadratic time complexity will restrict the maximum feasible number of markers. To overcome this, we reduce the number of possible positions (i) in the MERGE and POLISH based on how different the SCORE can be for adjacent positions. As there are only m individuals, it is plausible that $O(m)$ marker positions suffices. Moreover, as typically $n >> m$, this yields significant speedup, and at best gives a total runtime of $O(m^2 n \log n)$. By reducing the possible positions to those of having absolute difference in the SCORE probabilities ≤ 0.01, we notice at least up to 5x speedups. This difference limit can be controlled by the parameter identicalLimit in LM3.

Finally, the MERGE and POLISH are implemented to utilise multiple cores in parallel. In our experiments, we obtain 3-10x speedup on using 4-32 cores (data not shown).

2.4 Scaling to more markers

The computation burden of OrderMarkers2 (of LM3) scales easily to 100,000s of markers per chromosome. However, the number of differently segregating markers in a chromosome is limited by the number of individuals in a cross (assuming map lengths are $\leq$ 100cM). The marker ordering becomes ill-posed when the number of markers is much higher than the number of individuals as the clustering of identical markers becomes most important factor in the likelihood of the order. Especially in the whole genome sequencing data, there can easily be 1000 times more markers than individuals. One solution would be to use artificially small recombination parameters but choosing the suitable parameters can be tricky.

Instead, we have implemented natural data scaling procedure, controlled by parameter s, to LM3 to cope with this problem. In effect, this scales the genotype (log)likelihoods so that data likelihood would correspond to fraction s of the total markers. This can be seen as a continuous version of subsampling markers. However, this procedure (as well as subsampling) can miss the first recombinations at the map ends. To fix this, the recombination parameters are scaled as well on the map ends. The scale parameter is given to OrderMarkers2 as "scale=NUM1 NUM2", where NUM1 is the scale parameter (e.g. 0.001 if there are 1000 time more markers than individuals) and recombinations between NUM2/NUM1 first and last markers are scaled with fraction changing linearly from NUM1/NUM2 (first and last possible recombination) to 1 (no scaling).

2.5 Simulated data

To evaluate the performance of LM3 and some other software, data was simulated using custom scripts with varying rates and types of errors and missing genotypes. Phased bi-allelic double haploid (DH) data with 10,000 markers, 100 individuals and an average map length of 100 cM were simulated as follows.

The first 10 datasets (10k) were simulated by random errors with error rate of 1% without missing genotypes and recurrent recombinations were simulated with lower rate (recombination $r + 1$ occurred with probability of 1/10 of recombination r). The remaining 20 datasets were done similarly but with genotype likelihoods simulated according to five (10k-5x) or 10 (10k-10x) fold sequencing coverage for having homozygote and heterozygote genotypes as the two alleles and with read error rate of 1%. The first data was given to LM3 in its likelihood (called posterior) format where likelihoods gave the simulation error rate (flat 1%) and for MSTmap the possible erroneous genotypes were given as such. For the other datasets, the most likely genotypes were given to MSTmap if the error rate calculated from the likelihoods was lower than 1% or 5%, thus sometimes containing missing genotypes. All simulated data and scripts to generate them are provided with LM3.

For the new comparisons, we decided to concentrate only on performance of MSTmap and LM3, and for completeness, we run Lep-MAP2 on a subset of datasets as well. The linkage mapping results on these datasets and on the three programs are given in Table 2.

The parameters for LM3 were default, except scale=0.01 2 (100x more markers than individuals) and numMergeIterations=1 (faster runtime) and for 10k data $\theta_1 = 0.01$ (recombination1), $i_1 = 0.01$ (interference1) (making genotype likelihoods and recombination parameters equal). Note that these parameters are not the same used in the data simulation, e.g. recombination rate in the simulation was 1/10000 whereas default recombination parameter θ_1 (and θ_2) is 0.001 in LM3. Lep-MAP2 parameters were default except for minError=0.01 and phasedData=1. For MSTmap, we tried different parameter combinations. Parameter "detect_bad_data yes" was used as it did shorten the maps considerably.

2.6 Real data

We also constructed *de novo* linkage map for *Heliconius erato* from Illumina whole genome sequencing data of 93 offspring and their parents. The used sequencing data can be downloaded from SRA database with accession number SRP081917. This data was used in the genome assembly of *H. erato* (Van Belleghem *et al.*, 2017) and during the assembly process LM3 (marker separation) was developed. The parents were sequenced with 30-40x coverage and offspring with 7-12x coverage. We mapped raw reads to *H. erato* genome using BWA mem (Li, 2013) and using LM3 pipeline (pileupParser.awk, pileup2posterior.awk) we constructed the genotype likelihoods (posteriors) from the output of SAMtools mpileup (Li *et al.*, 2009). Only SNP variants were called. The used reference sequence can be obtained from Lepbase (Challis *et al.*, 2016). Only the ParentCall2 module was run on the likelihoods to obtain the final data (the Filtering2 module was not used nor needed).

First SeparateIdenticals was run on the data with LOD score limit 26.5 on the paternally informative markers only. Only markers occurring at least 4 times were kept and artificial data with about 24000 markers was generated based on the segregation patterns of these markers. Then all real markers were added to these artificial markers using JoinSingles2Identicals module and LOD score limit 25. The artificial marker data was run on SeparateChromosomes2 to find 21 chromosomes and mapping from identical markers to chromosomes. Then the chromosome assignment with 2.98 million markers was created using simple scripting from the output files of LM3.

Then OrderMarkers2 was run given the chromosome assignment with parameters $i_1 = \theta_1 = 10^{-6}$ and numMergeIterations=1, scale=0.002 3 and minError=0.01. The runtime for each chromosome was under 8 hours using at most 5 threads and required 20Gb of memory. The maps were almost exactly correct, however, the map end did have a handful of orphan markers that were removed as erroneous markers. The maps are visualised in Figure 4 and in the Supplement. We also tried MSTmap on this data, results of this experiment are described in the Supplement.

3 Results

The results on simulated data in Table 2 show that LM3 obtains better map accuracy (rank correlation up to 0.08 higher) than MSTMap. Also on poor quality data and large number of markers, LM3 can be up to 20x faster. Moreover, LM3 has only linear memory requirement unlike MSTmap whose memory requirement increases quadratically on the number of markers, making it unpractical on, roughly over 200k markers. LM3 also calculates the marker positions accurately being able to cluster identical markers together, whereas marker positions outputted by MSTmap can be several orders of magnitude inflated, making it unsuitable for contig orientation (see the Supplement for a real example). The results also show that LM3 outperforms Lep-MAP2 on each aspect, it is more accurate, constructs correct map lengths more accurately and is up to 2000x faster.

The LM3 performance is equally good on the real data. Only a single iteration on Algorithm 1 was sufficient on each chromosome to find the correct order and the only manual step involved removing some markers from the map ends (the erroneous markers are clearly visible from the Lep-MAP graph of Figure 4 and are very likely errors as these do not occur at the scaffold ends).

The obtained marker density is almost 1/100bp locating the recombinations precisely in the genome (which could be biologically interesting). The preliminary linkage maps on this data constructed with preliminary LM3 were successfully used in *Heliconius erato* project (Van Belleghem *et al.*, 2017) to obtain almost chromosome level assembly (scaffolds) as well as to detected contamination contigs. The map can also be used to find long structural variants, like inversions, as well as indels demonstrated in Figure 4.

	τ (accuracy)			time (seconds)			map length (cM)		
dataset	MST	LM2	LM3	MST	LM2	LM3	MST	LM2	LM3
10k	0.997	0.984*	**1.000**	6080	163000*	**361**	2960	663*	**106**
10k-10x	0.990	0.996*	**1.000**	**183**	605000*	310	793	113*	**96.2**
10k-5x	0.922	0.883*	**0.997**	872	293000*	**354**	9190	881*	**98.4**

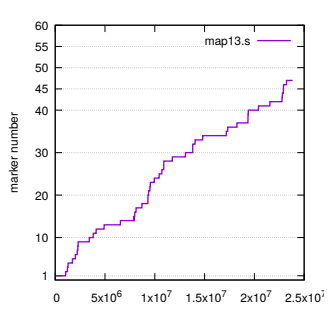
Table 2. Performance comparison of MSTmap (MST), Lep-MAP2 (LM2) and Lep-MAP3 (LM3) on simulated data of 10,000 markers and 100 individuals. Reported time is given in seconds on a Desktop computer with Intel Core Duo processor running at 3.6GHz using a single core. τ is the Kendall rank correlation (Kendall, 1938) of actual order and the constructed map order using only one marker among possible multiple markers at each actual map position. Map lengths are reported by the corresponding program and each value is an average on 10 independent datasets (* due to the high computing time of Lep-MAP2, it was run only on the first dataset (of 10)). The datasets were simulated with a simple model of recombination interference and with random error rate of 1% (10k) or with more realistic genotypic data obtained from simulated sequencing data with c fold coverage (10k-cx). See the main text for more info on the simulations and runs.

Moreover, some genetic regions did have strange markers. These seemed to be due to large, up to 1Mb, indels on some offspring (which may be quite common in this species or in this cross). Some maps are illustrated in Figure 4 and the sequencing coverage for one such problematic region. However, by easy manual inspection, these can be detected (and corrected if needed). The correlation of the physical length and the map length for *H. erato* chromosomes is given in the Figure 5.

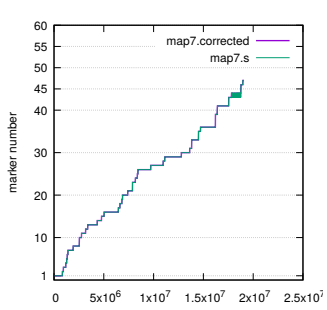
4 Discussion and Conclusion

Lep-MAP3 is the only tool suitable for mapping both, millions of markers and low-coverage sequencing data. This combination has high potential for genome assemblies but also for many other analyses. Millions of markers build strong evidence for, e.g. recombinations or lack of them and for structural variants. Especially the genome assemblies can be scaffolded with high confidence given such maps. The use of low-coverage data makes linkage mapping more practical to a wide range of non-model species, e.g. species with very long genomes or poor yield of DNA.

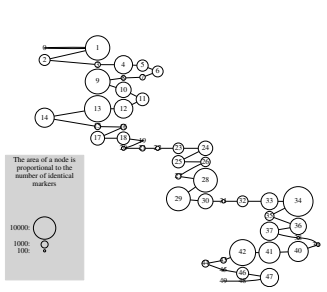
As future work, we would like to automate the integration of genome assemblies and linkage maps, possibly during the assembly process, and to further improve the speed and accuracy of LM3.



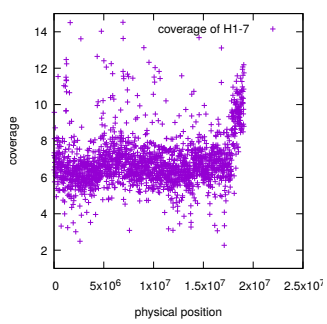
(a) Chr 13



(b) Chr 7



(c) Chr 13



(d) Chr 7

Fig. 4: Linkage maps for *H. erato* chromosomes 13 and 7, both having over 200k markers (and these chromosomes were assembled into single scaffolds). In figures a) and b), maps are made more clear by plotting the median marker number (rank in the order) for each window of 10kb containing at least 10 markers. Figure c) (Lep-MAP graph) shows the segregation patterns outputted by OrderMarkers2, closest patterns are joined by an edge, the numbers correspond to y-coordinates of a) (the patterns 0, 48 and 49 have been removed). For chromosome 7, the coverage for individual H1-7 is plotted in figure d). The peak in coverage at the end seems to be a duplication and is causing the error in the map (marker number alters between 43 and 44). The marker number is the the rank order in which the markers occur in the result. Note: the centiMorgan distances can be obtained by multiplying the marker number by 100/93 (100/number of individuals).

Acknowledgements

We thank John Davey, Leena Salmela, Petri Kemppainen and Virpi Ahola for useful comments and Owen McMillan, A. Tapia, M. Vargas and C. Rosales for providing and generating the used data.

Funding

The author has been funded by the European Research Council (grant 339873 to Chris Jiggins) as well as the Academy of Finland (grant 1292737 to Juha Merilä).

Conflict of Interest: none declared.

References

Ahola, V., Lehtonen, R., Somervuo, P., and *et al.* (2014). The Glanville fritillary genome retains an ancient karyotype and reveals selective chromosomal fusions in Lepidoptera. *Nature Communications*, **5**, 4737.

Catchen, J. (2015). Chromonomer. Available online. Accessed: 2016-08-19.

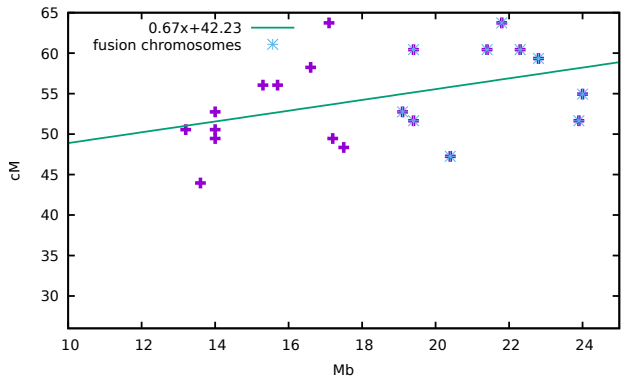


Fig. 5: Physical length and map length for all chromosomes for *H. erato*. In Lepidoptera, 31 (1n) is the ancestral chromosome number, whereas in *Heliconius*, 10 of these ancestral chromosomes have been fused (Ahola *et al.*, 2014). All the fused chromosomes are among the physically longest ones.

Challis, R. J., Kumar, S., Dasmahapatra, K. K. K., Jiggins, C. D., and Blaxter, M. (2016). Lepbase: the lepidopteran genome database. *bioRxiv*.

Cheema, J. and Dicks, J. (2009). Computational approaches and software tools for genetic linkage map estimation in plants. *Brief. Bioinform.*, **10**(6), 595–608.

Doerge, R. (2002). Mapping and analysis of quantitative trait loci in experimental populations. *Nat. Rev. Genet.*, **3**(1), 43–52.

Fierst, J. (2015). Using linkage maps to correct and scaffold de novo genome assemblies: methods, challenges, and computational tools. *Frontiers in Genetics*, **6**(220).

Kendall, M. G. (1938). A new measure of rank correlation. *Biometrika*, **30**(1-2), 81–93.

Laird, N. and Lange, C. (2008). Family-based methods for linkage and association analysis. In *Genetic Dissection of Complex Traits*, volume 60 of *Advances in Genetics*, pages 219 – 252. Academic Press.

Lander, E. and Green, P. (1987). Construction of multilocus genetic linkage maps in humans. *Proc. Natl. Acad. Sci.*, **84**(8), 2363–2367.

Li, H. (2013). Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *ArXiv e-prints*.

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., and Durbin, R. (2009). The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, **25**(16), 2078–2079.

Liu, D., Ma, C., Hong, W., Huang, L., Liu, M., Liu, H., Zeng, H., Deng, D., Xin, H., Song, J., Xu, C., Sun, X., Hou, X., Wang, X., and Zheng, H. (2014). Construction and analysis of high-density linkage map using high-throughput sequencing data. *PLoS ONE*, **9**(6), 1–9.

Morton, N. (1955). Sequential tests for the detection of linkage. *Am. J. Hum. Gen.*, **7**(3), 277–318.

Paterson, T. and Law, A. (2013). Arkmap: integrating genomic maps across species and data sources. *BMC Bioinformatics*, **14**(1), 1–10.

Rastas, P., Paulin, L., Hanski, I., Lehtonen, R., and Auvinen, P. (2013). Lep-map: fast and accurate linkage map construction for large snp datasets. *Bioinformatics*, **29**(24), 3128–3134.

Rastas, P., Calboli, F. C. F., Guo, B., Shikano, T., and Merilä, J. (2016). Construction of ultra-dense linkage maps with Lep-MAP2: stickleback F2 recombinant crosses as an example. *Genome Biology and Evolution*, **8**(1).

Simpson, J. T. and Pop, M. (2015). The theory and practice of genome sequence assembly. *Annual Review of Genomics and Human Genetics*, **16**(1), 153–172.

Van Belleghem, S., Rastas, P., and *et al.* (2017). Complex modular architecture around a simple toolkit of wing pattern genes. *Nature Ecology & Evolution*, **1**.

Van Ooijen, J. (2011). Multipoint maximum likelihood mapping in a full-sib family of an outbreeding species. *Genetics Research*, **93**, 343–349.

Wu, Y., Bhat, P. R., Close, T. J., and Lonardi, S. (2008). Efficient and accurate construction of genetic linkage maps from the minimum spanning tree of a graph. *PLoS Genet*, **4**(10), e1000212.